Lab2 实验报告

代码：

1. 数据进行了压位，并且输入是 8bit

```
#define N 1024
#define iB 16
#define jB 16
#define kB 16

typedef ap_uint<8> in_t;
typedef ap_uint<32> out_t;

typedef struct{
    in_t x[iB];
}pack_a_t;
typedef struct{
    in_t x[jB];
}pack_b_t;
typedef struct{
    out_t x[jB];
}pack_c_t;

typedef hls::axis<pack_a_t, 0, 0, 0> axis_in_a_t;
typedef hls::axis<pack_b_t, 0, 0, 0> axis_in_b_t;
typedef hls::axis<pack_c_t, 0, 0, 0> axis_out_t;
```

2. 整体框架是：N/iB 的大循环在 python 中，N/jB 的中循环代表一个 acc_C[iB][jB]算好了进行传回，N/kB 的小循环需要 load 数据和累加到 acc_C 上

```
    constexpr int I = N / jB;
    constexpr int J = N / kB;
    in_t buf_A[iB][kB], buf_B[jB][kB];
    out_t acc_C[iB][jB];
    for(int i = 0; i < iB; i++)
        for(int j = 0; j < jB; j++)
            acc_C[i][j] = 0;

    for(int i = 0; i < I; i++){
        for(int j = 0; j < J; j++){
#pragma HLS dataflow
            load<typename hls::stream<axis_in_a_t>, iB, kB>(in_A, buf_A);
            load<typename hls::stream<axis_in_b_t>, jB, kB>(in_B, buf_B);
            compute(buf_A, buf_B, acc_C);
        }
        store(acc_C, out_C, (i+1) == I);
    }
```

3. load 就正常一个个赋值，compute 同时进行 256 个计算

```
void compute(in_t buf_A[iB][kB], in_t buf_B[jB][kB], out_t acc_C[iB][jB]){
#pragma HLS array_partition variable=buf_A dim=2 cyclic factor=16
#pragma HLS array_partition variable=buf_B dim=2 cyclic factor=16
#pragma HLS array_partition variable=acc_C dim=2 cyclic factor=16
    for(int i = 0; i < iB; i++)
#pragma HLS pipeline II=1
        for(int j = 0; j < jB; j++){
#pragma HLS unroll
            for(int k = 0; k < 16; k++)
                acc_C[i][j] += buf_A[i][k] * buf_B[j][k];
        }
}
```

4. store 也正常一个个赋值，最后标记 last 信号

```
void store(out_t acc_C[iB][jB], hls::stream<axis_out_t> &out_C, bool last){
    for(int i = 0; i < iB; i++){
        axis_out_t x;
        for(int j = 0; j < jB; j++){
            x.data.x[j] = acc_C[i][j];
            acc_C[i][j] = 0;
        }
        x.last = last && (i+1==iB);
        x.keep = -1;
        out_C.write(x);
    }
}
```

结果：

```
.............................................................
FPGA time: 14.87987756729126
CPU time: 177.0373466014862
Sample: (984:997, 127:144)
Validate: True
Speedup: 11.897769037471601
Score: 100
```